WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Sunday, July 02, 2006

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L72	l68 and (virus adj definition and command$4 and restor$5)	14
☐	L71	l68 and (virus adj definition same command$4 same restor$5)	0
☐	L70	l68 and (virus adj definition same command$4 same restor)	0
☐	L69	((virus or viral or malicious)and (restor$5 and (data adj file or database)and command)).clm.	9
	DB=PGPB; PLUR=YES; OP=OR		
☐	L68	((virus or viral or malicious)and (restor$5 and (data adj file or database)and command))	1145
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L67	5442699.pn.	2
☐	L66	((MALIVANCHUK near2 TARAS) or (DARZI near2 MOSHE) or (ROTSCHIELD near2 OFER)) and restor$7	3
☐	L65	((MALIVANCHUK near2 TARAS) or (DARZI near2 MOSHE) or (ROTSCHIELD near2 OFER)) and (virus or malicious or viral or infected) and command	2
☐	L64	6519703.pn.	2
☐	L63	((MALIVANCHUK near2 TARAS) or (DARZI near2 MOSHE) or (ROTSCHIELD near2 OFER)) and (virus or malicious or viral or infected)	6
☐	L62	(L60 or l59 or l58 or l57) and ((restor$7 near5 (system or computer$2)) same command$4)	21
☐	L61	(L60 or l59 or l58 or l57) and (restor$7 same command$4)	83
☐	L60	380/$.ccls. and ((virus or viral or malicious)and (restor$5 and (data adj file or database)and command))	71
☐	L59	709/$.ccls. and ((virus or viral or malicious)and (restor$5 and (data adj file or database)and command))	247
☐	L58	726/$.ccls. and ((virus or viral or malicious)and (restor$5 and (data adj file or database)and command))	173
☐	L57	713/$.ccls. and ((virus or viral or malicious)and (restor$5 and (data adj file or database)and command))	229
☐	L56	713/$.ccls. and ((virus or viral or malicious)and (restor$5 and (data adj file or database)and command)).clm.	3
☐	L55	((virus or viral or malicious)and (restor$5 and (data adj file or database)and command)).clm.	9
☐	L54	((virus or viral or malicious)and (restor$5 and (data adj file or database)and command)).clm	1679
☐	L53	((virus or viral or malicious)and (restor$5 and (data adj file or database)and comand)).clm	0
☐	L52	((virus or viral or malicious)and (restor$5 and (data adj file or databse)and comand)).clm	0
☐	L51	L48 and (restor$7 same (registry adj file$2 or system adj file$2))	5
☐	L50	L48 and (restor$7 near3 (registry adj file$2 or system adj file$2))	3

☐	L49	L46 and (modify$3 near$2 registry adj file$2)	497
☐	L48	L47 and (malicous or intrusion or infect$7)	550
☐	L47	L46 and ((restor$8 near4 registry)or(system near 2 file$2))	923
☐	L46	(726/22 \|726/24 \|713/188).ccls.	1060
☐	L45	714/38 and malicious adj code	33
☐	L44	714.ccls. and virus	0
☐	L43	(709/224).ccls. and (virus or malicious adj code)	256
☐	L42	(713/201 \|713/188).ccls. and (virus or malicious adj code)	262
☐	L41	(713/201 \|713/188).ccls. and (virus or malicious adj code)	262
☐	L40	L39 and (dat adj file$2) and read and write and delete and rename	0
☐	L39	(713/200).ccls. and (virus or viral or malicious)	1
☐	L38	(5,408,642 5,349,655 5,613,002).pn.	6
☐	L37	5408642.pn.	2
☐	L36	L35 and read and write and delete	3
☐	L35	5408642 and windows	14
☐	L34	6401210	10
	DB=USPT; PLUR=YES; OP=OR		
☐	L33	5485575.pn.	1
☐	L32	5485575.pn.	1
☐	L31	5613002.pn.	1
☐	L30	5623600.pn.	1
☐	L29	5918008.pn.	1
☐	L28	6067410.pn.	1
☐	L27	6067410.pn.	1
☐	L26	6108799.pn.	1
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L25	L24 and rename	26
☐	L24	L23 and write	92
☐	L23	L22 and read	176
☐	L22	virus near4 file$4 same (command$4 or action$2)	230
☐	L21	L19 and virus and delete and rename	29
☐	L20	L19 and virus	87
☐	L19	L17 and recovery	262
☐	L18	L17 and recovery	262
☐	L17	system adj file$4 same windows	1254
☐	L16	L13 and windows adj 98	9
☐	L15	L13 and wsock$5	0
☐	L14	L13 and wsock32	0

☐	L13	system adj file$4 same virus	245
☐	L12	L11 AND VIRUS	52
☐	L11	(717/124 \|717/125 \|717/126 \|717/127 \|717/128).CCLS.	1882
☐	L10	L9 AND COMMAND$2	111
☐	L9	L8 SAME COMPUTER	262
☐	L8	VIRUS SAME DATA ADJ FILES	436
☐	L7	VIRUS SAME DATA ADJ FILE	436
☐	L6	L4 and (command same restor$7)	28
☐	L5	L4 same command same restor$7	4
☐	L4	(((data near3 file) or database) near5 (virus or malicious))	947
☐	L3	5657445.pn. and virus	1
☐	L2	5657445.pn.	2
☐	L1	20020178375	2

END OF SEARCH HISTORY

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
" restoring computer system" + "malicious code"
SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **restoring computer system malicious code** Found **106** of **178,880**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 81 - 100 of 106 Result page: previous 1 2 3 4 **5** 6 next
Relevance scale

81 Security: Methods for the prevention, detection and removal of software security vulnerabilities
Jay-Evan J. Tevis, John A. Hamilton
April 2004 **Proceedings of the 42nd annual Southeast regional conference**
Publisher: ACM Press
Full text available: pdf(424.09 KB) Additional Information: full citation, abstract, references, index terms

Over the past decade, the need to build secure software has become a dominant goal in software development. Consequently, software researchers and practitioners have identified ways that malicious users can exploit software and how developers can fix the vulnerabilities. They have also built a variety of source code security checking software applications to partially automate the task of performing a security analysis of a program. Although great advances have been made in this area, the core p ...

Keywords: functional programming, static code analysis

82 Emerging technologies for homeland security: Cyber defense: art to science
O. Sami Saydjari
March 2004 **Communications of the ACM**, Volume 47 Issue 3
Publisher: ACM Press
Full text available: pdf(108.20 KB) html(28.44 KB) Additional Information: full citation, abstract, references, index terms, review

Seeking the knowledge and means to more methodically detect, defend against, and better understand attacks on networked computer resources.

83 A basic security plan for a generic organization
Bhagyavati, Glenn Hicks
October 2003 **Journal of Computing Sciences in Colleges**, Volume 19 Issue 1
Publisher: Consortium for Computing Sciences in Colleges
Full text available: pdf(40.79 KB) Additional Information: full citation, abstract, references, index terms

Different organizations have plans in place to implement security in their computers and networks. In this paper, we examine the pros and cons of implementing a generic security plan on wired networks. The work presented here stems from an Independent Study course taken by an undergraduate student in the spring 2003 semester. It is hoped that this paper will provide new ideas for educators to direct student research in this important

subject as Independent Study or Directed Research courses.We f ...

84 Intrusion detection: Countering code-injection attacks with instruction-set randomization

Gaurav S. Kc, Angelos D. Keromytis, Vassilis Prevelakis

October 2003 **Proceedings of the 10th ACM conference on Computer and communications security**

Publisher: ACM Press

Full text available: pdf(146.35 KB) Additional Information: full citation, abstract, references, citings, index terms

We describe a new, general approach for safeguarding systems against *any* type of code-injection attack. We apply Kerckhoff's principle, by creating process-specific randomized instruction sets (*e.g.,* machine instructions) of the system executing potentially vulnerable software. An attacker who does not know the key to the randomization algorithm will inject code that is invalid for that randomized processor, causing a runtime exception. To determine the difficulty of integrating su ...

Keywords: buffer overflows, emulators, interpreters

85 Safely executing untrusted code: Model-carrying code: a practical approach for safe execution of untrusted applications

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Publisher: ACM Press

Full text available: pdf(301.30 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a new approach called *model-carrying code* (MCC) for safe execution of untrusted code. At the heart of MCC is the idea that untrusted code comes equipped with a concise high-level model of its security-relevant behavior. This model helps bridge the gap between high-level security policies and low-level binary code, thereby enabling analyses which would otherwise be impractical. For instance, users can use a fully automated verification procedure to determine if the code ...

Keywords: mobile code security, policy enforcement, sand-boxing, security policies

86 Migration: Luna: a flexible Java protection system

Chris Hawblitzel, Thorsten von Eicken

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Publisher: ACM Press

Full text available: pdf(1.39 MB) Additional Information: full citation, abstract, references

Extensible Java systems face a difficult trade-off between sharing and protection. On one hand, Java's ability to run different protection domains in a single virtual machine enables domains to share data easily and communicate without address space switches. On the other hand, unrestricted sharing blurs the boundaries between protection domains, making it difficult to terminate domains and enforce restrictions on resource usage. Existing solutions to these problems restrict sharing in an ad-hoc ...

87 Session 9: traffic analysis: Code-Red: a case study on the spread and victims of an internet worm

David Moore, Colleen Shannon, k claffy

November 2002 **Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment**
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, citings, index terms

On July 19, 2001, more than 359,000 computers connected to the Internet were infected with the Code-Red (CRv2) worm in less than 14 hours. The cost of this epidemic, including subsequent strains of Code-Red, is estimated to be in excess of $2.6 billion. Despite the global damage caused by this attack, there have been few serious attempts to characterize the spread of the worm, partly due to the challenge of collecting global information about worms. Using a technique that enables global detectio ...

Keywords: Code-Red, Code-RedI, CodeRedI, CodeRedII, backscatter, epidemiology, security, virus, worm

88 Type-Safe linking with recursive DLLs and shared libraries
Dominic Duggan
November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 6
Publisher: ACM Press
Full text available: pdf(658.62 KB) Additional Information: full citation, abstract, references, citings, index terms

Component-based programming is an increasingly prevalent theme in software development, motivating the need for expressive and safe module interconnection languages. Dynamic linking is an important requirement for module interconnection languages, as exemplified by dynamic link libraries (DLLs) and class loaders in operating systems and Java, respectively. A semantics is given for a type-safe module interconnection language that supports shared libraries and dynamic linking, as well as circular ...

Keywords: Dynamic Linking, Module Interconnection Languages, Recursive Modules, Shared Libraries

89 Security of runtime extensible virtual environments
Ernesto J. Sallés, James Bret Michael, Michael Capps, Don McGregor, Andrzej Kapolka
September 2002 **Proceedings of the 4th international conference on Collaborative virtual environments**
Publisher: ACM Press
Full text available: pdf(128.78 KB) Additional Information: full citation, abstract, references, index terms

Distributed, real-time virtual environment (VE) architectures have traditionally been driven by quality of service (QOS) considerations, with little or no concern paid to security issues. With recent advancements in functionality, computing power and network bandwidth it has become practical to use VEs in sensitive areas such as product development with proprietary information and visualization of classified information. Consequently, previously ignored aspects of security need to be made a prim ...

Keywords: collaborative environment, computer security, information assurance, runtime extensible virtual environment, security, taxonomy, virtual environment, virtual reality

90 Security for Web Applications and P2P: Abstracting application-level web security
David Scott, Richard Sharp

May 2002 **Proceedings of the 11th international conference on World Wide Web**
Publisher: ACM Press

Full text available: pdf(287.51 KB) Additional Information: full citation, abstract, references, citings, index terms

Application-level web security refers to vulnerabilities inherent in the code of a web-application itself (irrespective of the technologies in which it is implemented or the security of the web-server/back-end database on which it is built). In the last few months application-level vulnerabilities have been exploited with serious consequences: hackers have tricked e-commerce sites into shipping goods for no charge, user-names and passwords have been harvested and condential information (such as ...

Keywords: application-Level web security, component-based design, security policy description language

91 Security service level agreements: quantifiable security for the enterprise?
Ronda R. Henning
September 1999 **Proceedings of the 1999 workshop on New security paradigms**
Publisher: ACM Press
Full text available: pdf(662.02 KB) Additional Information: full citation, references, citings, index terms

92 From system F to typed assembly language
Greg Morrisett, David Walker, Karl Crary, Neal Glew
May 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 3
Publisher: ACM Press

Full text available: pdf(483.91 KB) Additional Information: full citation, abstract, references, citings, index terms

We motivate the design of typed assembly language (TAL) and present a type-preserving ttranslation from Systemn F to TAL. The typed assembly language we pressent is based on a conventional RISC assembly language, but its static type sytem provides support for enforcing high-level language abstratctions, such as closures, tuples, and user-defined abstract data types. The type system ensures that well-typed programs cannot violatet these abstractionsl In addition, the typing constructs admit ...

Keywords: certified code, closure conversion, secure extensible systems, type-directed compilation, typed assembly language, typed intermediate languages

93 Resource bound certification
Karl Crary, Stephnie Weirich
January 2000 **Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**
Publisher: ACM Press

Full text available: pdf(1.71 MB) Additional Information: full citation, abstract, references, citings, index terms

Various code certification systems allow the certification and static verification of important safety properties such as memory and control-flow safety. These systems are valuable tools for verifying that untrusted and potentially malicious code is safe before execution. However, one important safety property that is not usually included is that programs adhere to specific bounds on resource consumption, such as running time. We present a decidable type system capable of specify ...

94 Inoculating software for survivability

Anup K. Ghosh, Jeffrey M. Voas
July 1999 **Communications of the ACM**, Volume 42 Issue 7
Publisher: ACM Press
Full text available: pdf(214.10 KB) html(37.50 KB) Additional Information: full citation, references, citings, index terms

95 "Mainstreaming" automated information systems security engineering (a case study in security run amok)

James W. Coyne, Norman C. Kluksdahl
November 1994 **Proceedings of the 2nd ACM Conference on Computer and communications security**
Publisher: ACM Press
Full text available: pdf(750.61 KB) Additional Information: full citation, abstract, references, index terms

Since mid-1992, the National Aeronautics and Space Administration (NASA) Mission Control Center (MCC) in Houston has pursued commonality and consolidation of facilities and equipment to reduce the implementation costs associated with the modernization of the control center, as well as the on-going operations and sustaining costs. In December of 1993, a management team, comprised almost entirely of individuals with manned spaceflight operations experience, was given the charter of completing ...

96 Workshop on architectural support for security and anti-virus (WASSA): Energy-security tradeoff in a secure cache architecture against buffer overflow attacks

Koji Inoue
March 2005 **ACM SIGARCH Computer Architecture News**, Volume 33 Issue 1
Publisher: ACM Press
Full text available: pdf(391.20 KB) Additional Information: full citation, abstract, references, index terms

In this paper, we propose a cache architecture, called *SCache*, to detect buffer-overflow attacks at run time. Furthermore, the energy-security efficiency of SCache is discussed. SCache generates replica cache lines on each return-address store, and compares the original value loaded from the memory stack to the replica one on the corresponding return-address load. The number and the placement policy of the replica line strongly affect both energy and vulnerability. In our evaluation, it i ...

97 Academic papers: Information security curriculum creation: a case study

Bradley Bogolea, Kay Wijekumar
October 2004 **Proceedings of the 1st annual conference on Information security curriculum development**
Publisher: ACM Press
Full text available: pdf(89.42 KB) Additional Information: full citation, abstract, references, index terms

Information Security is a critical part of the technology infrastructure. A survey of undergraduate degree programs in Computer Science, Information Technology, Management Information Science, and others show a lack of emphasis on security issues in their curriculum. The purpose of this paper is to present a case study on our approach to creating an undergraduate curriculum that will enhance existing degree programs in Computer Science and Information Technology to provide an increased awareness ...

Keywords: assessment, curriculum, information assurance, information security, information technology, infosec, security, surveys

98 Trustworthy systems: Symmetric behavior-based trust: a new paradigm for internet computing

Vivek Haldar, Michael Franz
September 2004 **Proceedings of the 2004 workshop on New security paradigms**
Publisher: ACM Press
Full text available: pdf(76.86 KB) Additional Information: full citation, abstract, references

Current models of Internet Computing are highly asymmetric - a host protects itself from malicious mobile Java programs, but there is no way to get assurances about the behavior of a program running remotely. The asymmetry stems from a behavior-based security model: hosts ensure conformance to a given security policy by restricting the actions of programs. In contrast, security models that are based on cryptography (including code signing) are inherently symmetric by design but do not match the ...

99 Databases: SQL DOM: compile time checking of dynamic SQL statements

Russell A. McClure, Ingolf H. Krüger
May 2005 **Proceedings of the 27th international conference on Software engineering**
Publisher: ACM Press
Full text available: pdf(353.48 KB) Additional Information: full citation, abstract, references, index terms

Most object oriented applications that involve persistent data interact with a relational database. The most common interaction mechanism is a call level interface (CLI) such as ODBC or JDBC. While there are many advantages to using a CLI -- expressive power and performance being two of the most key -- there are also drawbacks. Applications communicate through a CLI by constructing strings that contain SQL statements. These SQL statements are only checked for correctness at runtime, tend to be f ...

Keywords: SQL, SQL DOM, SQL injection, SQL strings, dynamic SQL, impedance mismatch

100 Opinion: Electronic voting systems: the good, the bad, and the stupid

Barbara Simons
October 2004 **Queue**, Volume 2 Issue 7
Publisher: ACM Press
Full text available: pdf(594.04 KB) html(25.97 KB) Additional Information: full citation, index terms

Results 81 - 100 of 106 Result page: previous 1 2 3 4 **5** 6 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player